

**FIG. 1**  
(Prior Art)

2/7

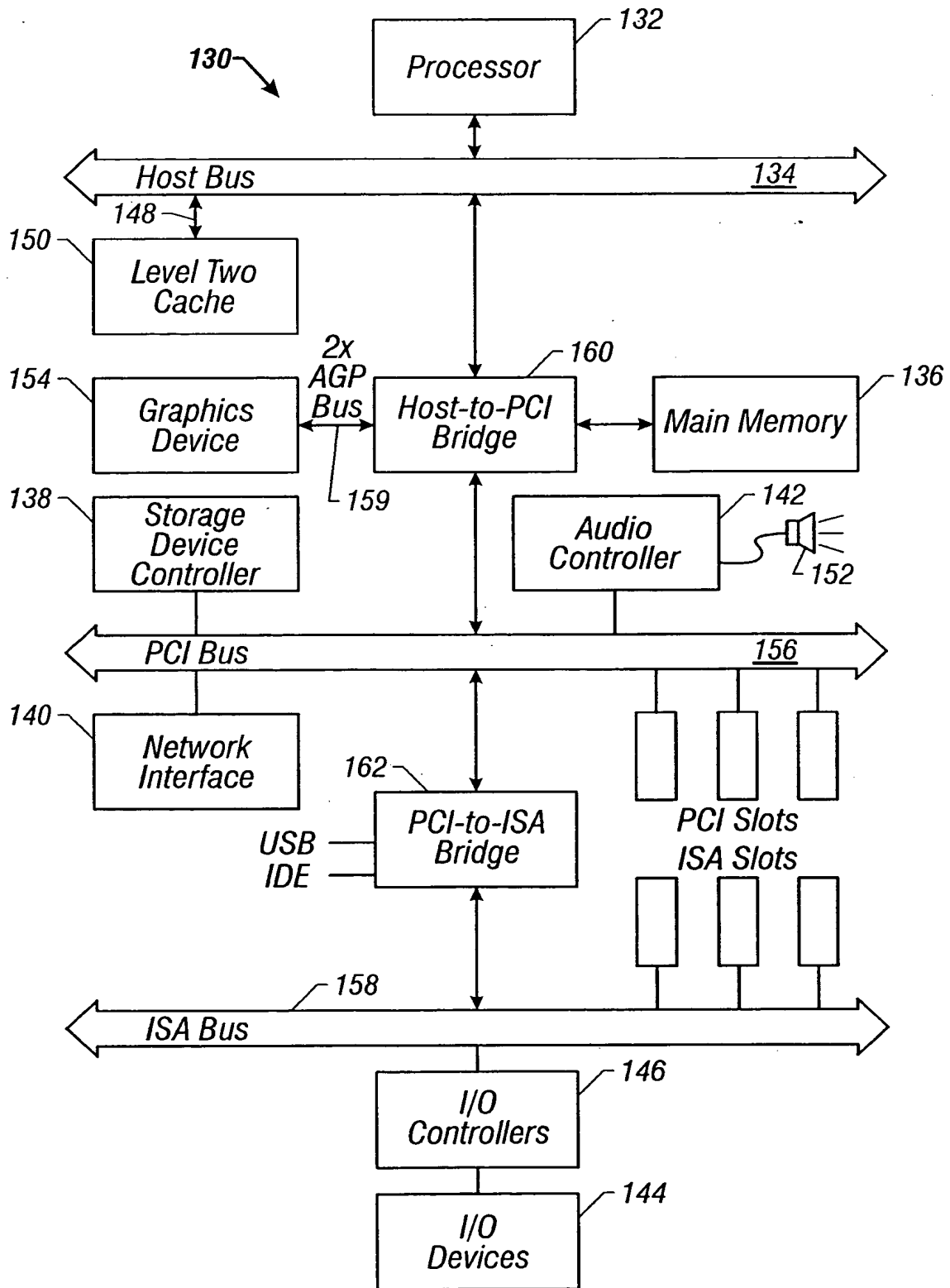


FIG. 1A

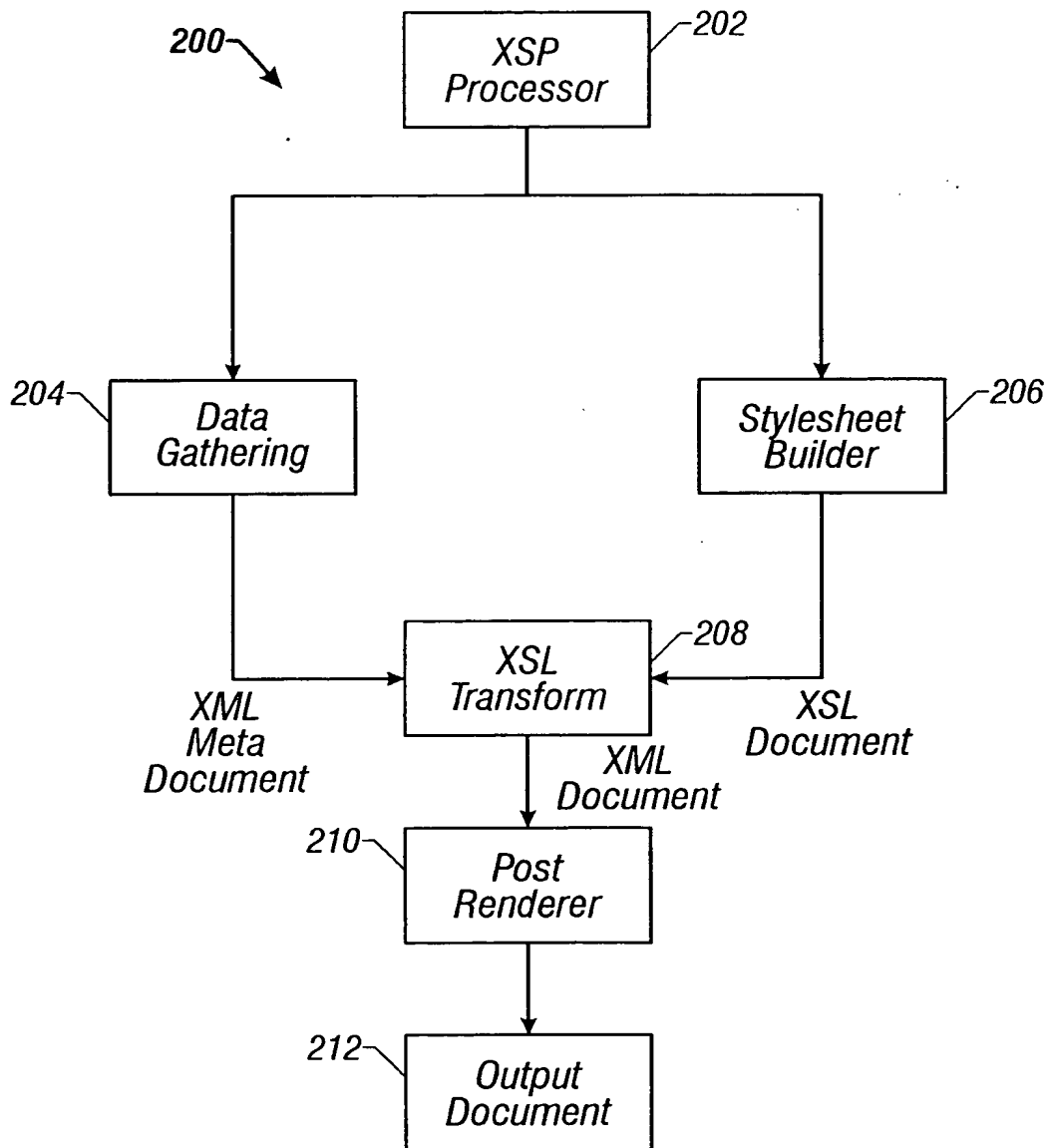


FIG. 2

300

FILE = inventors.xsp

```

302 <xsp: script base="inventor-base.xsp" xmlns:xsp="uri:xsp" xmlns:xsl="uri:xsl"
    xmlns:doc=uri:doc">
    <!-- This script extends inventor-base.xsp -->
    <!-- Over-ride contents implementation in the base script -->
    <xsp:sub name="contents">
304     <!-- Ask XSP to get the list of inventors as a server-side data island
        and hold onto it for script lifetime -->
306     <xsp:xml name="listofinventors" src="/xsp/data/inventors/xsp.xml" />
        <!-- Use XSP to query the data island, with a query attribute to
308     retrieve only the inventors (not managers) -->
        <xsp: query src="#listofinventors" query="// inventors">
            <!-- standard XSL root rule to start stylesheet processing or
                this document fragment -->
            <xsl:template match="/">
                <!--XSL formatting statements go here-->
            </xsl:template>

            <!-- XSP to call base subroutine to import the default rules -->
            <xsp:call href="#core-templates" />

            <!-- XSL again to provide a special template for an inventor,
                outputting selected fields in the desired order -->
            <xsl:template match="inventor">
                <!-- XSL formatting statements go here-->
            </xsl:template>
        </xsp:query>

        <!-- Use XSP to query the same data island a second time for
            managers and not inventors -->
        <xsp:query src="#listofinventors" query="//managers">
            <xsl: template match="/">
                <-- Put statements for standards XSL stylesheet
                    processing for this portion of the document -->
            <xsl:template>
306     </xsp:query>
308 </xsp:sub>
    </xsp:script>

```

FIG. 3

FILE = inventor - base.xsp 400

```

402 <xsp:script xmlns:xsp="uri:xsp" xmlns:xsl="uri:xsl" xmlns:doc="uri:doc">
    <!-- Base script -->

    <!-- Default XSP entry point -->
    <xsp:sub name="main">
404     <!-- Insert any standardised output here -->
        <!-- Then call the contents routine -->
406     <xsp:call href="#contents" />
408 </xsp:sub>

    <!-- Provide a default XSP contents routine -->
    <xsp:sub name="contents">
410     <!-- Ask XSP to query the document -->
        <xsp:query src="/xsp/data/inventors/xsp.xml">
            <!-- Now XSL begins - just provide a default rule -->
            <xsl:template match="/">

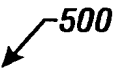
                ....

                </xsl:template>
                <!-- Now call an XSP routine to import the default rules before
                    invoking XSL -->
428     <xsp:call href="#core-templates"
        </xsp:query>
432 </xsp:sub>

434 <!-- This XSP routine provides two XSL default rules whenever needed -->
    <xsp:sub name="core-templates">
        <!-- output all text nodes as is -->
        <xsl:template match="textNode()">
            <xsl:value-of />
        </xsl:template>
        <!-- ensure XSL recursion continues in the absence of better
            matching rules -->
        <xsl:template match="*">
            <xsl:apply-templates />
        </xsl:template>
452 </xsp:sub>
454 </xsp:script>

```

FIG. 4

FILE = xsp.xml 

```

<?xml version="1.0" encoding="iso-8859-1"?>
<inventorlist>
  <managers>
    <manager role="dir">
      <name>Gordon Ballantyne </name>
      <title>Director of EMBA Online </title>
    </manager>
    <manager role="vp">
      <name>Maurice Cowey </name>
      <title>Vice President of HSB for EMBA </title>
    </manager>
  </managers>
  <inventors>
    <inventor ID="I001">
      <name>David Brooke </name>
      <title>Internet Development Manager </title>
      <SSN/WL 08 77 96 A </SSN>
      <region>EMEA </region>
      <badgeno>64329 </badgeno>
    </inventor>
    <inventor ID="I002">
      <name>Steve Saxon </name>
      <title>Technical Architect </title>
      <SSN>NH 77 71 96 B </SSN>
      <region>EMEA </region>
      <badgeno>103836 </badgeno>
    </inventor>
  </inventors>
</inventorlist>

```

FIG. 5

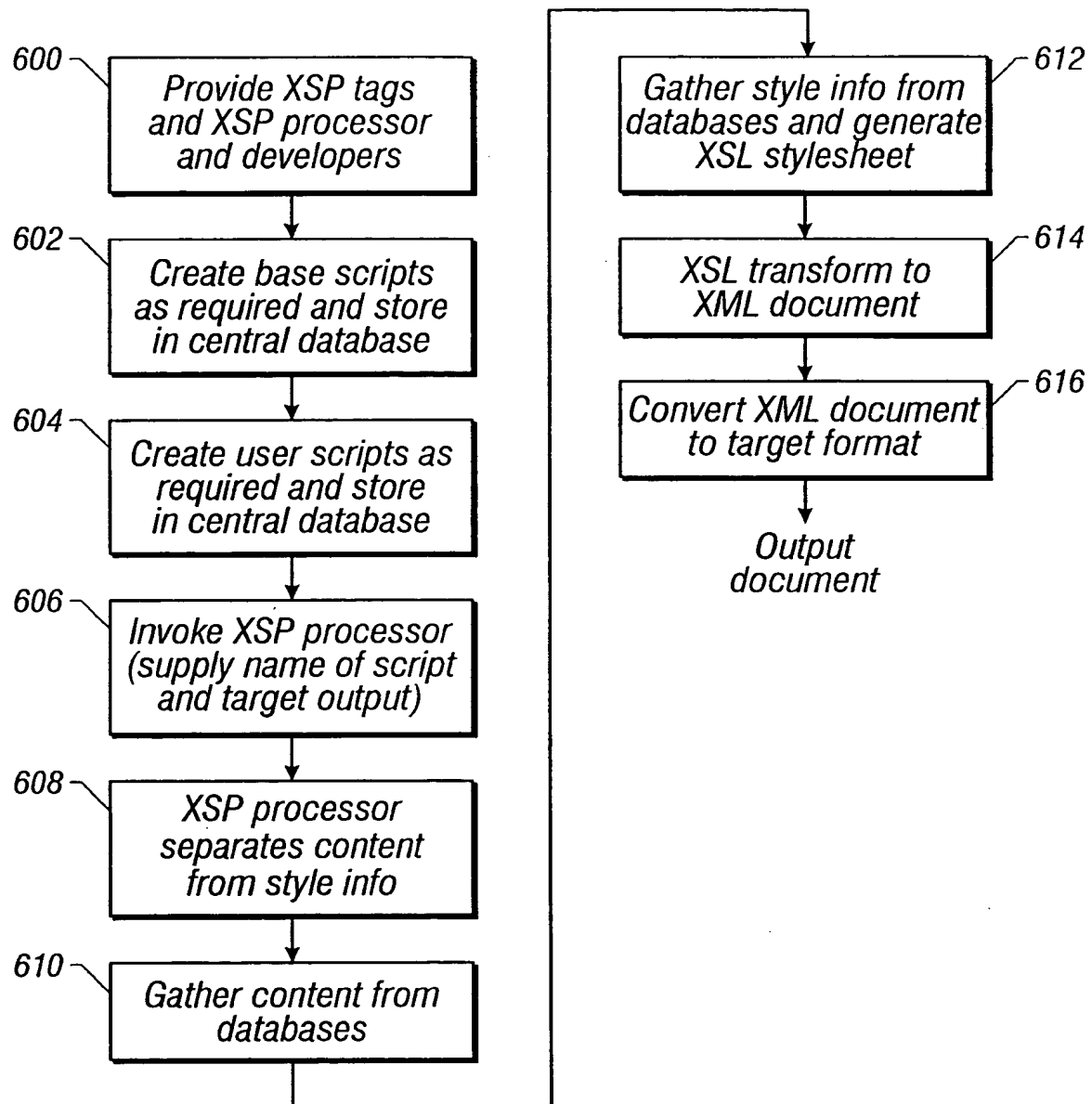


FIG. 6